VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Vývoj nástroje pro analýzu nezabezpečeného provozu v bezdrátových Wi-Fi sítích

Development of a Tool for Analyzing Unsecured Traffic in Wireless Wi-Fi Networks

2020 Matěj Stuchlík

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Matěj Stuchlík**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R059 Mobilní technologie

Téma: Vývoj nástroje pro analýzu nezabezpečeného provozu v bezdrátových Wi-Fi sítích
Development of a Tool for Analyzing Unsecured Traffic in Wireless Wi-Fi Networks

Jazyk vypracování: čeština

Zásady pro vypracování:

V současné době je otázka bezpečnosti mobilních zařízení a přenosu dat ve Wi-Fi sítích stále častěji diskutované téma. Všichni výrobci mobilních zařízení a vývojáři aplikací se snaží zabezpečit přenášený obsah různými způsoby. V případě špatného návrhu aplikace nebo systému, může být přenos nešifrován a v případě zachycení takovéhoto provozu potenciálním útočníkem, může tak dojít k vážným bezpečnostním incidentům. Hlavním cílem bakalářské práce je zpracování následujících bodů zadání:

1. Analýza zachyceného provozu v bezdrátových Wi-Fi sítích.
2. Podrobná analýza možných bezpečnostních rizik.
3. Vývoj a optimalizace nástroje pro analýzu nezabezpečeného provozu na třetí a čtvrté vrstvě OSI modelu.
4. Vývoj nástroje pro notifikaci v případě, že bude rozpoznán nezabezpečený provoz v síti.
5. Testování a vyhodnocení navrženého řešení.

Seznam doporučené odborné literatury:

[1] RAHALKAR, Sagar. *Network Vulnerability Assessment: Identify security loopholes in your network's infrastructure*. Birmingham: Packt Publishing, 2018. ISBN 978-1788831925.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Lukáš Kapičák**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky ID16 OSTRAVA

prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. května 2020 ..... Matěj Gluchý

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 15. května 2020

Matěj Hudík

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Lukáši Kapičákovi za konzultace a cenné rady při návrhu nástroje, dále také za rady při psaní bakalářské práce.

Abstrakt

Tato bakalářská práce se zabývá vytvořením nástroje pro analýzu síťového provozu a notifikaci uživatelů, kteří při své činnosti na internetu využijí nezabezpečených webových služeb. Nástroj dokáže upozornit uživatele na nezabezpečený provoz do několika vteřin. Ke správnému fungování je vytvořen přístupový bod na zařízení Raspberry Pi 4 Model B, který podporuje autokonfiguraci pomocí protokolu DHCP. Uživatelé, kteří chtějí využít internetových služeb přístupového bodu a být upozorněni na případné nebezpečí se musí nejprve autorizovat prostřednictvím Captive Portalu, kde je nutno zadat uživatelské jméno a email.

Klíčová slova: Raspberry Pi; Wi-Fi; Captive Portal; přístupový bod; Linux

Abstract

This bachelor thesis is about development of a tool for analyzing network traffic and user notification, for those who use unsecured web services while being on the internet. The tool can alert the user to insecure traffic within a few seconds. For the tool to work properly, an access point is created on the Raspberry Pi 4 Model B, which supports autoconfiguration via DHCP protocol. Users, who want to use the Internet services of the access point and be notified of possible dangers, must first authorize themselves through the Captive Portal, where it is necessary to enter a username and email.

Key Words: Raspberry Pi; Wi-Fi; Captive Portal; Access Point; Linux

Obsah

Seznam použitých zkratek a symbolů

DHCP	–	Dynamic Host Configuration Protocol, automatická konfigurace počítačů připojených do počítačové sítě
DNS	–	Domain Name System, hierarchický, decentralizovaný systém doménových jmen
DSSS	–	Direct-sequence spread spectrum, technika rozdělení spektra kde jsou originální data vynásobena náhodným kódem pro šíření rušení
FHSS	–	Frequency-hopping spread spectrum, metoda vysílání rádiových signálů s využitím rychlé změny nosné frekvence
GIT	–	distribuovaný systém správy verzí
HTTP	–	Hypertext Transfer Protocol, internetový protokol pro přenost hypertextových dokumentů
IEEE	–	Institute of Electrical and Electronics Engineers, světová organizace pro pokrok v technologii
IOT	–	Internet Of Things, internet věcí
IP	–	Internet Protocol, protokol pracující na síťové vrstvě v počítačových sítích a internetu
IP adresa	–	jednoznačně identifikuje síťové rozhraní v počítačové síti, která používá IP protokol
ISM	–	Industrial, scientific and medical networks, průmyslové, vědecké a lékařské sítě
LAN	–	Local Area Network, lokální síť
MAC	–	Media Access Control, jednoznačný identifikátor síťového zařízení
MIMO	–	Multiple-Input Multiple-Output, vícenásobný vstup a vícenásobný výstup
OFDM	–	Orthogonal Frequency Division Multiplexing, ortogonální multiplex s frekvenčním dělením
OS	–	Operating System, Operační systém
OSI	–	Open Systems Interconnection, referenční model komunikace na síti
PAN	–	Personal Area Network, osobní síť na krátké vzdálenosti
QAM	–	Quadrature amplitude modulation, kvadraturní amplitudová modulace
TCP/IP	–	Transmission Control Protocol/Internet Protocol, primární přenosový protokol/protokol síťové vrstvy
TLS	–	Transport Layer Security, protokol zabývající se šifrováním přenosu dat

WEP	–	Wired Equivalent Privacy, zastaralý bezpečnostní standard bezdrátových sítí
WiFi	–	označení standardů IEEE 802.11 popisujících bezdrátovou komunikaci v počítačových sítích
WLAN	–	Wireless Local Area Network, bezdtrátová lokální síť
WMAN	–	Wireless Metropolitan Access Network, bezdtrátová metropolitní síť
WPA	–	Wi-Fi Protected Access, modernější zabezpečení bezdrátových sítí
WPAN	–	Wireless Personal Area Network, bezdrátová osobní síť
WWAN	–	Wireless Wide Area Network, rozsáhlá bezdrátová síť

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Tato bakalářská práce se zabývala vývojem nástroje pro analýzu nezabezpečeného provozu v bezdrátových Wi-Fi sítích. Jedna z motivací pro tuto práci bylo zaměřit se na bezpečností rizika, jenž vznikají používáním nezabezpečené Wi-Fi sítě nebo nezabezpečených internetových služeb.

Hlavním tématem bakalářské práce je vývoj nástroje, který zachytává provoz na nezabezpečené Wi-Fi síti a následně uživatele informuje prostřednictvím emailu o možném narušení jejich bezpečnosti vlivem úniku citlivých dat. Tento nástroj tak zvyšuje ochranu uživatelů při využití nezabezpečených webových služeb.

Dalším tématem této práce je popis bezdrátových sítí a základních vlastností standardů IEEE 802.11. Poté se práce zabývá principem komunikace zařízení v internetových sítích, zabezpečení a získávání dat z této komunikace.

Nástroj pro zachytávání nezabezpečeného provozu byl vyvinut na zařízení Raspberry Pi 4 Model B a napsán v jazyce Python za použití příkazů v OS Linux.

Práce byla rozdělena na dvě části, kde v první části je popsána teorie potřebná pro pochopení principu získávání nezabezpečených dat, jenž je využit při vývoji nástroje v druhé části práce.

2 Bezdrátové sítě

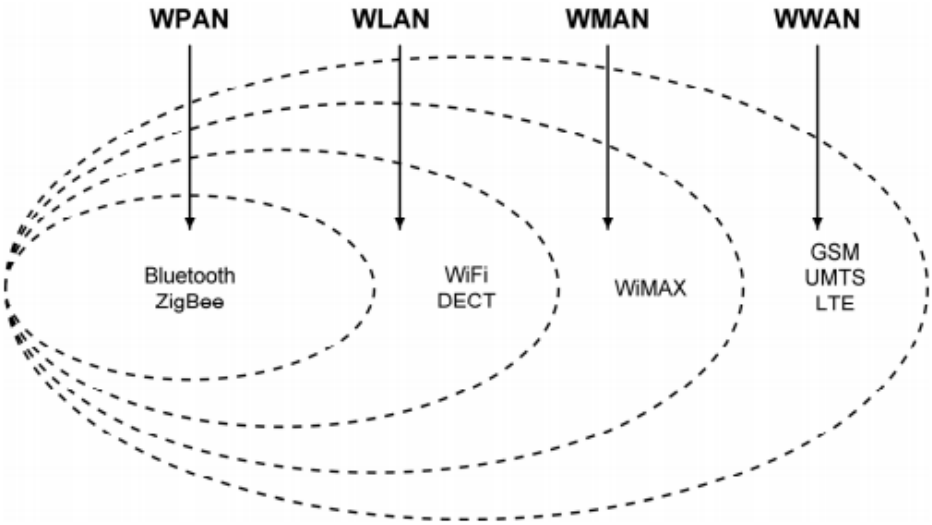
Bezdrátové sítě představují propojení zařízení uživatelů prostřednictvím rádiových vln. Mezi hlavní výhody bezdrátových sítí patří možnost rychle a jednoduše vytvářet sítě bez nutnosti kabeláže. Provádět výstavby počítačových sítí i na místech, pro které je z ekonomického hlediska nevýhodné natažení kabelů nebo na základě rozhodnutí památkového úřadu je v těchto místech výstavba kabelových sítí zakázána. Bezdrátové sítě mohou využívat i pohyblivé objekty [1].

2.1 Rozdělení bezdrátových sítí

Bezdrátové sítě slouží ke komunikaci mezi zařízeními a lze je rozdělit podle různých kritérií a parametrů [2].

Zde je popsáno rozdělení podle užití (dosahu, oblasti pokrytí signálem):

- WPAN (Wireless Personal Area Network) bezdrátové osobní sítě.
- WLAN (Wireless Local Area Network) bezdrátové lokální sítě.
- WMAN (Wireless Metropolitan Area Network) bezdrátové metropolitní sítě.
- WWAN (Wireless Wide Access Network) bezdrátové rozsáhlé sítě.



Obrázek 1: Rozdělení bezdrátových sítí podle rozsahu [2]

Dále tyto 4 typy sítí můžeme rozdělit do dvou kategorií dle svého dosahu – sítě krátkého dosahu SR (Short range) a sítě dlouhého dosahu LR (Long Range). Bezdrátové sítě s krátkým dosahem představují sítě na vymezeném prostoru s menší rozlohou, nejčastěji do desítek metrů. Tato definice se týká především bezdrátových sítí typu LAN, které nalezneme v administrativních

budovách, školních areálech, výrobních závodech ale i v domácnostech, avšak tato definice zahrnuje také bezdrátové sítě typu PAN, díky kterým mohou mezi sebou komunikovat přenosná zařízení, ale jen s výrazně omezeným dosahem, většinou se jedná o vzdálenosti v řádech desítek centimetrů až jednotek metrů.

Dostupné frekvence, na kterých jsou provozovány tyto sítě krátkého rozsahu jsou v různých zemích odlišné. Nejužívanějšími frekvenčními pásmy pro sítě typu WLAN (Wireless LAN) jsou pásma okolo 2,4 GHz a 5 GHz, která jsou volně dostupná ve většině zemí. Bakalářská práce je zaměřená zejména na sítě typu WLAN, protože k vytvoření takovéto sítě stačí použít hot-spot na mobilním telefonu či využít směrovač k vytvoření například domácí sítě.

3 Wi-Fi

Jedná se skupinu standardů 802.11, které pracují v bezlicenčních pásmech ISM. Tyto standardy spadají do kategorie WLAN, což představuje alternativu či rozšíření pro metalické LAN sítě [2].

3.1 Vývoj standardů 802.11

Samotný vývoj se týkal jak rychlosti přenosu dat, tak i nových technologií, které pracují na fyzické vrstvě, a také využití jiného pásma.

3.1.1 IEEE 802.11

Jednalo se o první standard s označením IEEE 802.11, který byl definován v roce 1997. Využívá pásmo 2,4 GHz až 2,485 GHz. Přenosová rychlost činila 1 nebo 2 Mbit/s. Fyzická vrstva využívala technologie FHSS nebo DSSS.

3.1.2 IEEE 802.11b

Standard 802.11b využívá stejné pásmo jako původní standard, avšak je schopen vyšších přenosových rychlostí. Dokáže dynamicky měnit přenosovou rychlost podle rušení. Maximální přenosová rychlost je 11 Mbit/s. Fyzická vrstva využívá oproti původnímu standardu výhradně technologii DSSS.

3.1.3 IEEE 802.11a

Největší novinkou tohoto standardu bylo využití pásma 5,15 až 5,725 GHz. Nabízí také až téměř 5tinásobnou rychlost přenosu oproti standardu 802.11b, a to až 54Mbit/s v závislosti na rušení. Nově fyzická vrstva využívá technologii OFDM.

3.1.4 IEEE 802.11g

Tento standard umožňuje stejnou rychlost přenosu jako 802.11a, tedy 54Mbit/s, avšak pracuje na pásmu 2,4 až 2,485 GHz. Standard využívá 64-QAM modulaci a princip OFDM nebo DSSS pro zpětnou kompatibilitu se standardem IEEE 802.11b.

3.1.5 IEEE 802.11n

Jedná se o první standard, který využívá obě pásma 2,4 až 2,485 GHz a 5,15 až 5,725 GHz. Maximální přenosová rychlost činní až 600 Mbit/s. Fyzická vrstva využívá principu OFDM a MIMO.

3.1.6 IEEE 802.11ac

Tento standard využívá pásmo 5GHz s adaptivní změnou šířky pásma. Přenosová rychlost až 1 Gbit/s. Používá se zde modulace až 256-QAM a 8x8 MIMO spolu s principem OFDM.

3.1.7 IEEE 802.11ad

Jako jediný standard využívá pásmo okolo 60GHz (v Evropě 57.00 - 66.00GHz). Má vysokou přenosovou rychlost až 7 Gbit/s. Využívá modulaci OFDM [26].

4 Získávání dat v bezdrátových sítích

Na získávání dat v bezdrátových sítích se lze dívat tak, že odposloucháváme komunikaci mezi dvěma či více zařízeními.

4.1 Základ komunikace

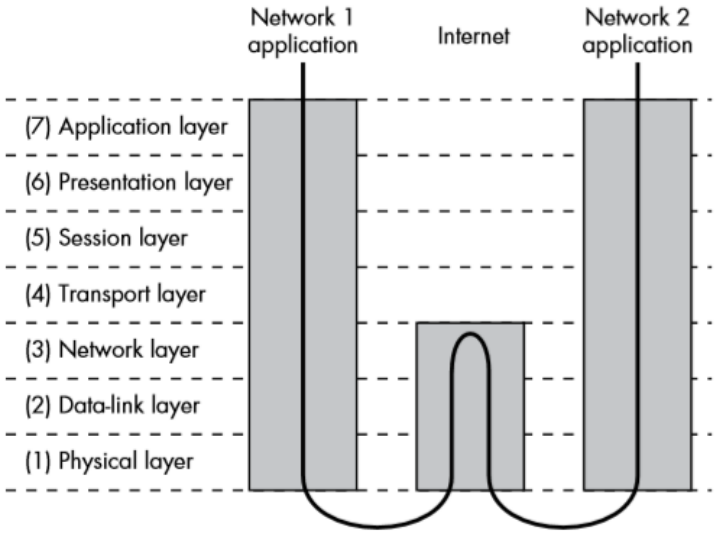
Základem komunikace lze rozumět předávání informací mezi dvěma a více zařízeními. Komunikace v síti probíhá zasíláním paketů, které obsahují dva druhy dat. Jedním typem jsou takzvaná řídící data, která slouží k ověření, zda byl paket doručen, a pokud ano, tak jestli nebyl nějakým způsobem poškozen. Druhým typem jsou uživatelská data, která obsahují informace, jež se paket snaží zaslat jinému zařízení.

4.2 OSI model

Aby spolu mohli zařízení komunikovat, musí se dorozumět společným jazykem. Tento jazyk je popsán strukturou vrstev v OSI modelu. Tento model obsahuje jakési standardy, které umožňují rozdělit komunikaci do více vrstev a hardware jako například směrovač je pak schopen se zaměřit na konkrétní část komunikace a ostatní ignorovat [4].

Struktura tohoto modelu vypadá následovně:

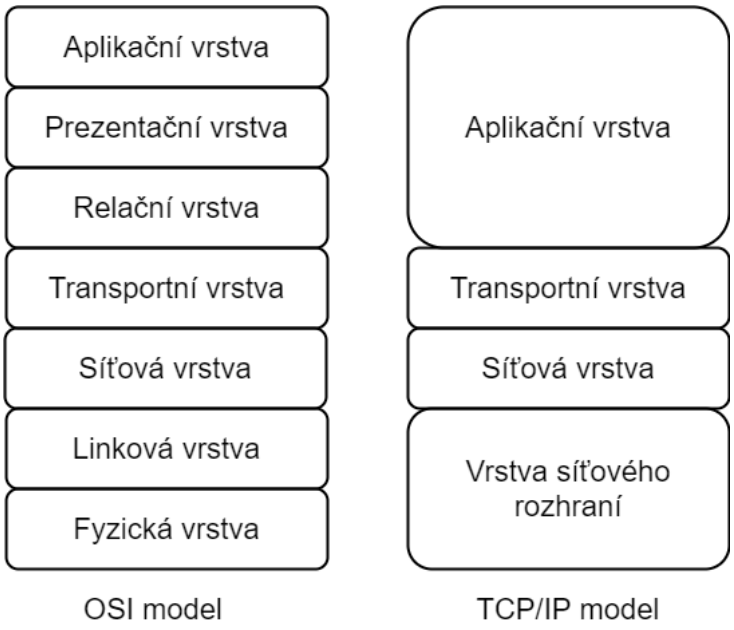
- Fyzická vrstva – Tato vrstva mimo jiné zajišťuje fyzické propojení mezi dvěma body, kteří spolu komunikují. Stará se o navazování i ukončování spojení a přenos čistých dat mezi zařízeními.
- Linková vrstva – Jde o vrstvu, která přenáší data mezi dvěma body. Na rozdíl od fyzické vrstvy nabízí také korekci chyb či možnost kontrolovat tok dat.
- Síťová vrstva – Poskytuje adresování a směrování mezi sítěmi. Slouží také jako prostředník pro spodní a horní vrstvy.
- Transportní vrstva – Umožňuje přenos dat mezi systémy. Poskytuje spolehlivou komunikaci dat a tím umožňuje vyšším vrstvám nezabývat se spolehlivostí přenosu dat.
- Relační vrstva – Je zodpovědná za navázání a údržbu spojení mezi síťovými aplikacemi.
- Prezentační vrstva – Stará se o prezentaci dat aplikacím v syntaxi nebo jazyku, kterým aplikace budou rozumět. Díky tomu se naskýtá mimo jiné použití funkcí jako například datová komprese a šifrování.
- Aplikační vrstva – Zabývá se sledováním požadavků aplikace. Jedná se o vrstvu, která má za účel poskytovat přístup aplikacím ke komunikačnímu systému.



Obrázek 2: Princip komunikace na internetu prostřednictvím OSI modelu [4]

4.3 TCP/IP model

TCP/IP Model byl navržen a vyvinut v 60. letech 20. století. Namísto sedmi vrstev, jak tomu bylo u OSI modelu, obsahuje pouze 4 vrstvy. Jedná se o více spolehlivou službu, založenou na základě protokolů, podle kterých pak vznikl samotný model TCP/IP. Jedná se tak o opačný přístup oproti OSI modelu, kde byl nejdříve navržen model a až později protokoly [5].



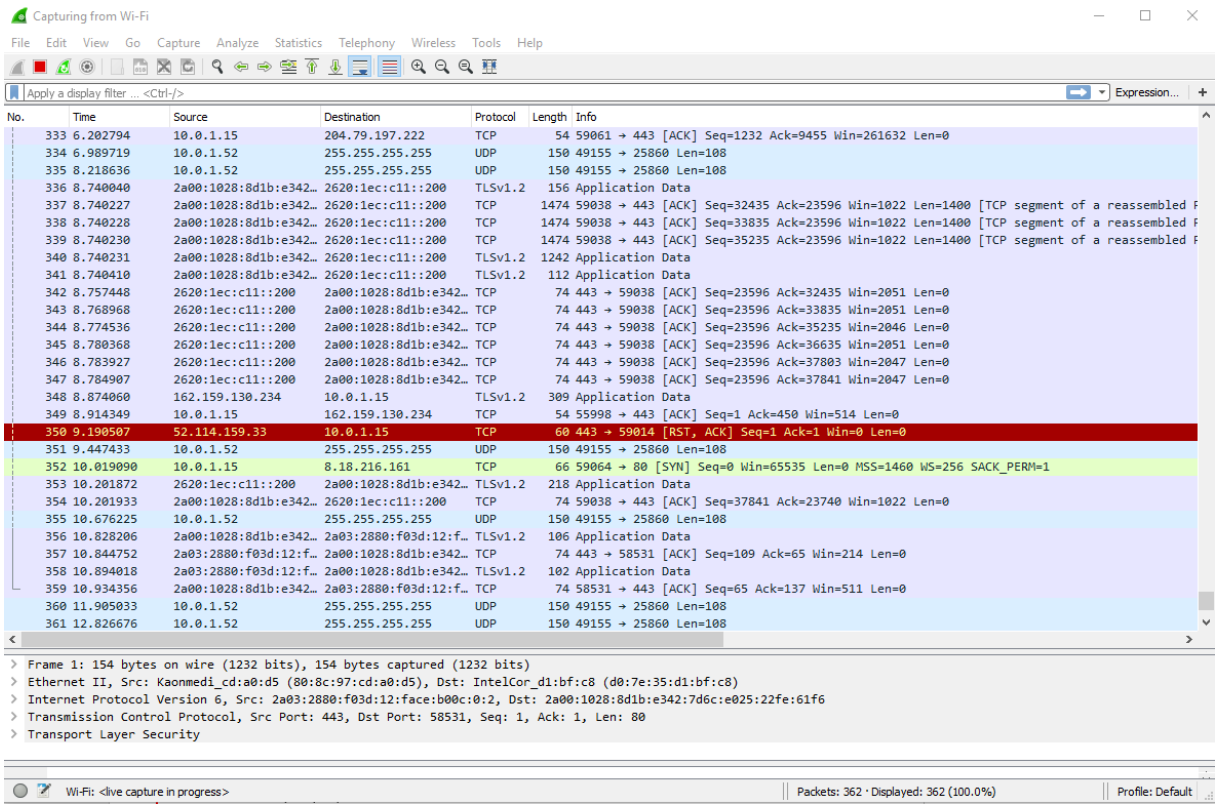
Obrázek 3: Porovnání OSI a TCP/IP modelu

4.4 Získávání dat

Získávání dat slouží k mnoha účelům v reálném světě, například pro zvýšení kvality služeb formou zpětné vazby od zákazníků nebo uživatelů. K získání dat, která si mezi sebou zasílají zařízení na síti, mohou sloužit různé programy, které dokážou tyto data (nejčastěji packety) filtrovat, zobrazit jejich obsah jako například zdrojovou i cílovou IP adresu, MAC adresu zařízení nebo porty webových služeb a usnadňují tak analýzu komunikace mezi dvěma zařízeními.

4.4.1 Wireshark

Tento program slouží k analýze síťových protokolů skrze grafické rozhraní, avšak je schopen pracovat také formou příkazů přes příkazovou řádku. Dokáže provádět analýzu sítě jak v reálném čase, kdy je možno si určit, které síťové rozhraní chceme sledovat, lze však mimo jiné Wireshark použít i k zobrazení již zachyceného provozu na síti, který byl uložen do souboru [6].



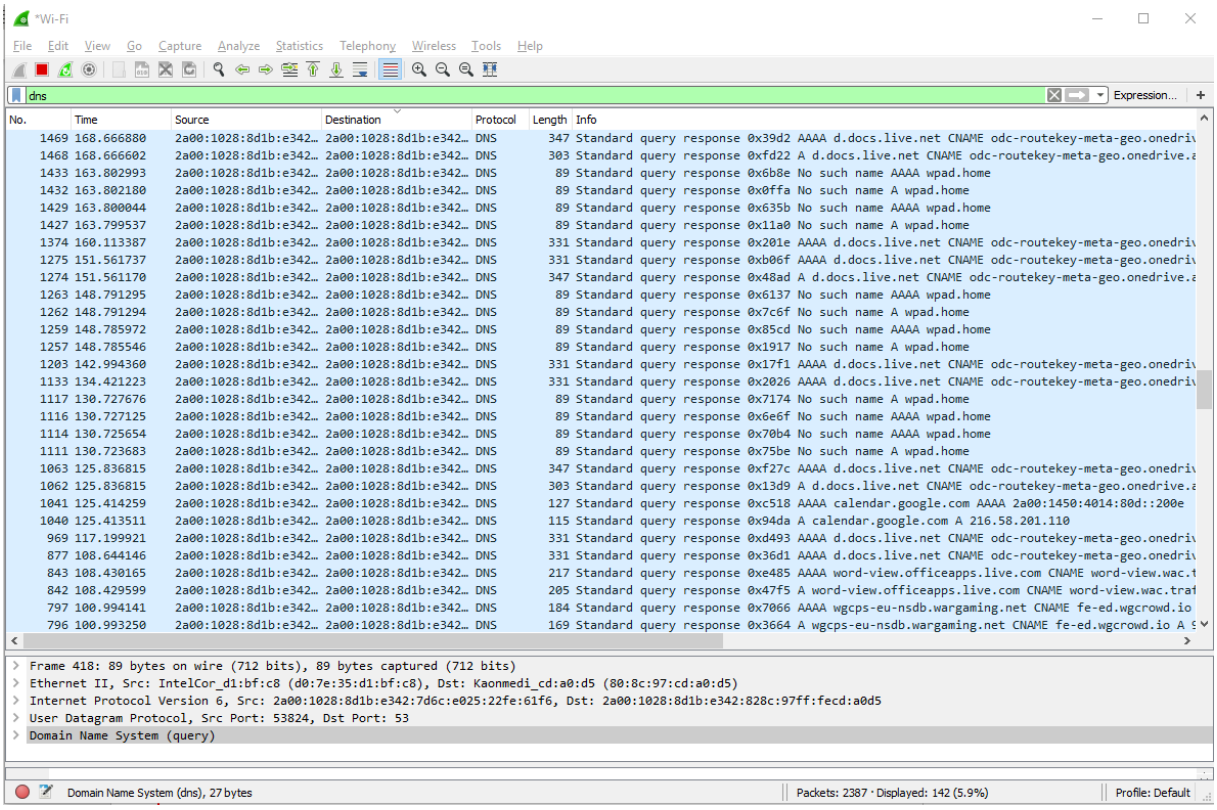
Obrázek 4: Zobrazení GUI Wiresharku při zachytávání reálného provozu

Grafické uživatelské rozhraní Wiresharku nabízí hned několik možností filtrace paketů, a to během záznamů sítě v reálném čase.

Zachycené pakety a jejich zobrazení lze filtrovat pomocí:

- čísla paketů (každý nový paket dostane pořadové číslo, které se automaticky zvyšuje)
- času v sekundách, kdy byl paket zachycen (počáteční čas je 0)
- zdrojové IP adresy
- IP adresy cíle, kterému je paket určen
- protokolu
- délky paketů
- informace, kterou paket nese

Nad tímto výpisem se nachází pole, pro vložení textového filtru. Takto lze filtrovat a zobrazit pakety například pouze určitého protokolu.



Obrázek 5: Zobrazení pouze protokolu DNS

4.4.2 TCPdump

Jedná se o program, který obdobně jako Wireshark slouží k analýze paketů, avšak formou příkazové řádky [7].

Když program TCPdump dokončí záznam paketů, vypíše počet:

- zachycených paketů
- paketů zachycených pomocí filtrů
- paketů zahozených kernelem (počet zahozených paketů kvůli nedostatku místa v bufferu)

Ukázka zachycení paketů pomocí TCPdumpu na rozhraní eth0:

```
#Příkaz pro spuštění TCPdumpu v terminálu
tcpdump -i eth0 -w test.pcap
#Oznamení, že program TCPdump začal zachytávat provoz na síti
tcpdump: listening on eth0, link-type EN10MB (Ethernet)
#Výpis po ukončení klávesovou zktratkou Ctrl+C
715 packets captured
727 packets received by filter
0 packets dropped by kernel
```

Výpis 1: Ukázka příkazu TCPdumpu

4.4.3 TShark

Jde o analyzátor síťových protokolů. Dokáže získávat data ze sítě v reálném čase, popřípadě dokáže přečíst a zobrazit pakety z dříve uloženého souboru, buď vypsáním dekódované formy těchto paketů uložených v souboru na standardní výstup, nebo tuto formu zapsat do souboru. Nativním formátem formátu pro zápis souborů se v TSharku používá formát pcapng, jenž využívá Wireshark a další podobné nástroje. Bez dalšího nastavení, program TShark pracuje podobně jako tcpdump. Používá knihovnu pcap pro zachytávání provozu a na standardním výstupu se pro každý přijatý paket zobrazí souhrnný řádek.

5 Monitoring síťového provozu

Monitoring sítě se využívá pro získávání dat z provozu na sítích. Takto získána data můžou sloužit jednak pro efektivnější využití sítě, avšak pokud dojde k monitoringu sítě neoprávněným uživatelem, mohou být tyto data zneužita ke špatným účelům (získání citlivých osobních údajů, narušení soukromí atd.).

Programy, které se využívají k monitorování sítí:

- Wireshark
- TCPdump
- Kismet

5.1 Monitorovací režim

Monitorovací režim RFMON (Radio Frequency MONitor), umožňuje počítači monitorovat veškerý provoz na bezdrátové síti. Tuto možnost však nabízí pouze za předpokladu, že síťová karta počítače dokáže přejít do monitorovacího režimu. V případě, že síťová karta na daném zařízení nepodporuje monitorovací režim, nabízí se možnost použití externích síťových adapterů, které monitorovací režim na zařízení umožní. Pro monitorování drátových LAN sítí se používá promiskuitní režim [10].

5.1.1 Využití monitorovacího režimu

Lze jej využít například k analýze packetů na bezdrátových sítích nebo ke sledování provozu a využití Wi-Fi sítí. Další možné využití je při návrhů nových bezdrátových sítí v určitém místě, protože je možno si pomocí monitorovací režimu zobrazit sítě, které jsou v určité oblasti a na určitém kanálu. Lze tak předejít zatížení sítě způsobené zahlcením kanálu, na kterém se Wi-Fi sítě nachází. Jedním z důležitých vlastností monitorovacího režimu je, že pro zachytávání packetů není třeba být připojen na žádný směrovač nebo síť.

5.1.2 Nastavení síťového adaptéru na monitorovací režim

Jednou z možností, jak nastavit síťový adaptér na monitorovací režim, je použití příkazu iwconfig. Tento příkaz se stará o nastavení bezdrátového síťového rozhraní [11].

Různé možnosti nastavení pomocí příkazu iwconfig:

- změna názvu sítě
- nastavení ID sítě
- změna režimu síťového rozhraní
- nastavení frekvence
- nastavení kanálu sítě

Další z možností, jak nastavit síťový adaptér na monitorovací režim je za pomocí programu Aircrack-ng. Tento program obsahuje celou řadu nástrojů pro práci se zabezpečením bezdrátové sítě nebo nástroje pro obcházení těchto zabezpečení. Dokáže testovat a popřípadě odstranit zabezpečení bezdrátových bezpečnostních protokolů WEP, WPA a WPA2. Jedná se o program, se kterým se pracuje prostřednictvím příkazové řádky, nejčastěji v terminálu OS Linux.

Několik nástrojů programu Aircrack-ng:

- Airmon-ng – Používá se pro nastavení módu síťových karet a k odstranění nepotřebných procesů při používání Aircrack-ng.
- Airodump-ng – Jedná se o takzvaný packet sniffer, který dokáže sledovat data od jednoho či více přístupových bodů. Využívá se pro analýzu přístupových bodů v okolí.
- Aircrack-ng – Jde o nástroj, který dokáže pomocí útoků na protokoly WPA/WPE získat klíč k přístupu na síť.

Nastavení monitorovacího režimu pomocí Aircrack-ng:

```
#Příkaz pro ukončení nežádoucích programů
sudo airmon-ng check kill
#Příkaz pro zapnutí monitorovacího režimu
sudo airmon-ng start [wireless interface]
```

Výpis 2: Ukázka nastavení monitorovacího režimu pomocí Aircrack-ng

5.2 Promiskuitní režim

Jako u monitorovacího režimu se jedná o označení pro režim síťové karty, avšak oproti monitorovacímu režimu se dá tento režim použít i na klasické drátové spojení. Většinou se tento mód využívá pro takzvaný packet sniffing, za účelem získání dat ze síťového provozu. Je velice obtížné najít v síti uživatele, který se snaží získat data, protože negenerují téměř žádný provoz.

Programy, které se používají pro packet sniffing:

- TCPdump
- Sniffit

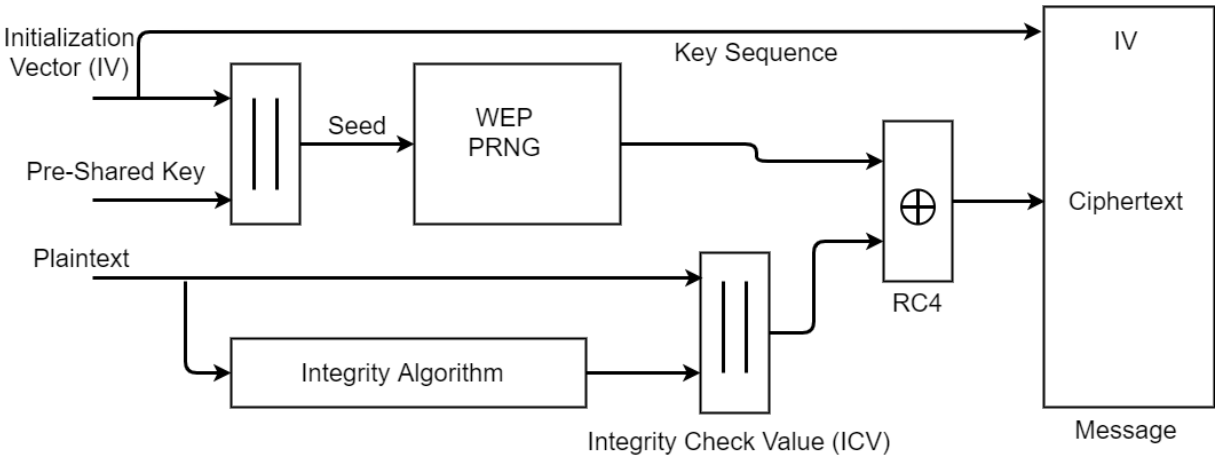
Dříve bylo nejhorší variantou pro administrátory sítí najít uživatele tam, kde je internet do počítačů přiveden skrze rozbočovač. Ten totiž vysílá data na všechny počítače v síti, a to i těm, pro které nejsou určena. V dnešní době se však rozbočovače nepoužívají, avšak existují útoky na přepínače, které se pak jako rozbočovače chovají. Počítače, kterým data nebyla určena, jednoduše tyto data zahodí, avšak uživatel, jenž chce jemu neurčená data získat, poruší toto pravidlo a přijme všechny data, která přepínač (nastaven jako rozbočovač) vyšle do sítě [12].

6 Šifrování a zabezpečení síťového provozu

Díky velkému počtu programů, které slouží k získávání dat ze síťového provozu, je velice snadné si zobrazit provoz na nezabezpečených sítích. Aby se předcházelo této potencionální hrozbě, vzniklo několik druhů zabezpečení. Slovem šifrování (encryption) označujeme kryptografický algoritmus, který převádí čitelnou zprávu (prostý text) na její nečitelnou podobu – šifrovaný text. Tento šifrovaný text je pro lidi nečitelný a díky tomu se zvyšuje úroveň bezpečnosti přenášených či uchovaných dat na počítači [13].

6.1 WEP

WEP (Wired Equivalent Privacy) byl navržen pro bezpečnost drátových sítí LAN za použití šifrování pomocí algoritmu RC-4. Nejdříve je vytvořen 40bitový klíč s 24bitovým inicializačním vektorem (IV), který slouží jako šifrovací a dešifrovací klíč. Novější verze WEP pak začali používat až 128bitový klíč. Výsledný klíč je pak základem pro pseudo-náhodný generátor čísel (PRNG). Dále se čistý text přenese do integračního algoritmu a připojí se znovu k čistému textu (Integrity Check Value - ICV). Potom ICV a výsledek klíčové sekvence projdou RC4 algoritmem. Výsledná šifrovaná zpráva je pak vytvořena přidáním IV před šifrovaný text.



Obrázek 6: Princip vytvoření šifrované komunikace [14]

WEP však patří mezi starší způsob ochrany a obsahuje řadu chyb, proto se dnes také používají modernější šifrovací verze bezpečnostních algoritmů.

6.2 WPA

Z důvodů mnoha nedostatků WEP byl vyvinut nový způsob ochrany, zaměřený hlavně na bezdrátové sítě. Z toho také plyne název Wi-Fi Protected Access (WPA). Zásadním rozdílem mezi WEP a WPA bylo použití 256bitového klíče pro každý paket [15]. Navíc obsahuje kontrolu integrity zpráv, takže nedochází k případné modifikaci a opětovnému zasílání datových packetů.

Díky této funkci se dokázalo zabránit útokům, které spočívaly v upravování a opětovné zaslání těchto upravených paketů.

Za použití integrity zpráv se pomocí matematické funkce zjistí případný rozdíl zaslaného paketu, jestliže tento výpočet neodpovídá správnému výpočtu, paket je označen jako zmanipulovaný a následně zahozen. WPA také užívá protokol TKIP (Temporal Key Integrity Protokol). Dalším rozdílem oproti WEP bylo dynamické vytváření klíčů jak pro uživatele, tak například i pro samotné pakety. Navíc tyto klíče byly oproti WEP distribuovány automaticky [16].

6.3 WPA2

WPA2 (WPA version 2) byl představen jako nový bezpečnostní standard 802.11i v roce 2004. Důležitou novinkou se stal nový standard šifrování AES (Advanced Encryption Standard), ten se později začal používat i pro WPA. Díky vylepšenému šifrování byl zde považován za největší bezpečnostní problém útočník, který již byl do sítě přihlášen, protože mohl napadnout ostatní zařízení na lokální síti.

Většina bezpečnostních opatření se týkala hlavně podnikových sítí, protože pro domácí sítě nebyly bezpečností hrozby příliš relevantní. Velikou slabinou se stala možnost připojení na přístupové body pomocí WPS (Wi-Fi Protected Setup), toto slabé místo zabezpečení bylo již známo ze sítí, které používaly WPA. Proto se doporučovalo možnost WPS na přístupových bodech vypnout [17].

6.4 Porovnání WPA a WPA2

Zásadní nevýhodou WPA2 byl potřebný výpočetní výkon na přístupových bodech, avšak s novým hardwarem se tento problém stal téměř irelevantní. WPA2 využívá modernější způsob ochrany přenosu dat a vyšší rychlostí šifrování dat, proto je doporučováno používat WPA2, pokud to směrovač umožní [17].

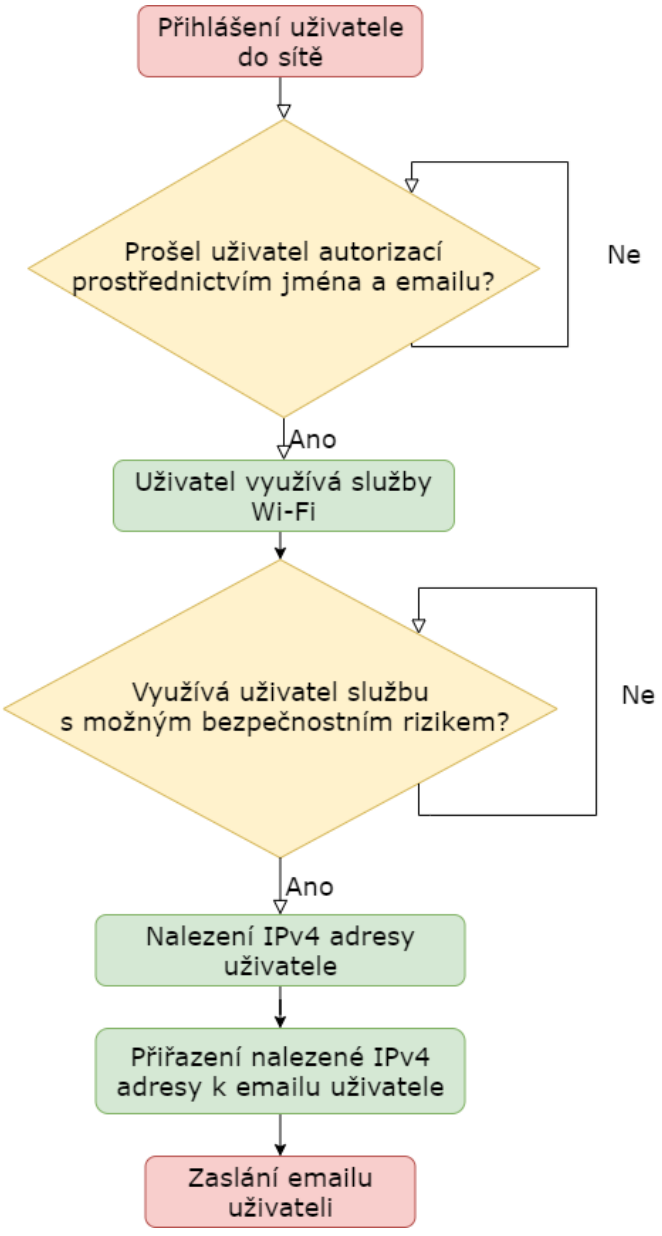
6.5 WPA3

Jedná se o nejmodernější způsob ochrany sítí. Jedním z hlavních vylepšení bezpečnosti je ochrana proti slovníkovým útokům, které znamenaly velký problém pro sítě používající WPA a WPA2. Slovníkový útok spočíval v hádání hesla, kdy se útočník snažil uhádnout heslo pomocí nejrůznější kombinací slovních spojení nebo samotných slov, protože WPA i WPA2 umožňovalo nekonečný počet pokusů pro zadání hesla [17].

WPA3 také nabízí snadnější připojení pro zařízení, které nemají vizuální konfiguraci rozhraní kvůli expanzi IoT a moderním přístrojům umožňující připojení k Wi-Fi [27].

7 Vývoj aplikace

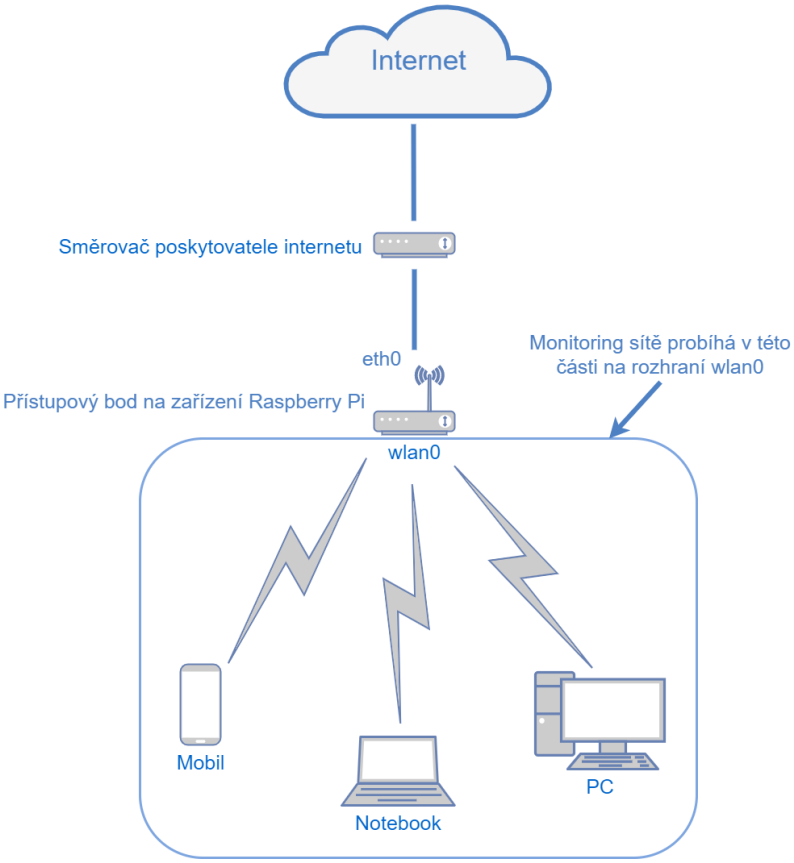
Aplikace by měla fungovat tak, že se prostřednictvím programu TCPdump spustí sledování sítě, kterou uživatel používá například k posílání e-mailů, přihlašuje se v této síti na webové stránky atp. Následně by měla aplikace průběžně kontrolovat, zda uživatel využívá nezabezpečené webové služby pro přenos dat či emailů. Pokud by k tomu došlo, uživatel by měl obdržet notifikaci prostřednictvím emailu o možném narušení bezpečnosti nebo odhalení svých osobních údajů.



Obrázek 7: Procesní diagram aplikace

7.1 Získání dat

V první fázi bylo třeba získat data ze sítě, která vzniknou provozem na této síti. Pro získání dat lze využít několik monitorovacích programů. Během vývoje aplikace jsem zvolil program TCPdump, díky kterému lze provádět monitoring skrze příkazovou řádku v terminálu OS Linux a uživatel tedy nebude nijak rušen během chodu aplikace. Jelikož program monitoruje celý provoz v síti, za použití správných příkazů lze docílit maximální možné míry sledování provozu a díky tomu zjistit, zda a popřípadě kde došlo k onomu narušení soukromí. Tento monitoring sítě probíhá na přístupovém bodě, který byl vytvořen na zařízení Raspberry Pi 4 model B.



Obrázek 8: Schéma domácí sítě

Následně bylo třeba zvolit, jak tyto příkazy využít ve vlastní aplikaci. Nejvhodnějším programovacím jazykem pro tuto práci byl Python. Hlavním důvodem byla efektivní implementace procesů OS Linux přímo v kódu aplikace, díky mnoha knihovnám, které tuto funkcionalitu umožňují [18].

Jako první tedy při implementaci aplikace bylo třeba vyřešit spouštění programu TCPdump uvnitř python souboru. První problém implementace nastal, když jsem se pokoušel spustit program, avšak kvůli omezení práv vyžaduje program TCPdump být spuštěn s vyššími právy.

Nastavení vyšších práv programu TCPdump:

```
sudo groupadd pcap
sudo usermod -a -G pcap $USER
sudo chgrp pcap /usr/sbin/tcpdump
sudo chmod 750 /usr/sbin/tcpdump
sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
sudo reboot
```

Výpis 3: Nastavení práv v příkazové řádce terminálu OS Linux

Pak už bylo vše v pořádku a python soubor zahájil monitoring bez chyby. Jako další tedy bylo nutné vyřešit monitoring tak, aby byl po určitém časovém úseku pozastaven a mohla být provedena kontrola souboru a zhodnocení případných bezpečnostních rizik

Jako první jsem vyzkoušel funkci, kterou TCPdump ovládá, tedy vložení do příkazu další argument, jenž způsobí to, že každých několik vteřin (přesný počet se nastaví číslem napsaným za zmíněným argumentem) TCPdump vytvoří nový soubor. Tento postup mi však přišel méně vhodný, protože by bylo nemožné zachytit dobu přesně tehdy, kdy se starý soubor uloží a program začne vytvářet soubor nový, protože se tato činnost provádí automaticky. Funkce automatického vytváření zachycených souborů tedy slouží hlavně k trvalému či dočasnému uchování záznamu sítě, například každou hodinu, kde není potřeba ji průběžně kontrolovat.

Bylo tedy nutné přejít k manuálnímu řešení uvedeného problému. Toho bylo docíleno za použitím uspání vlákna programu, ve kterém se příkaz provádí. Běh programu se sice na určitou dobu uspí, avšak díky tomu, že monitoring je spuštěn jako proces na pozadí, probíhá záznam sítě i během jeho uspání [19].

Toho můžeme využít v době, kdy chceme zjistit, zda došlo k nějakému bezpečnostnímu riziku, a tedy k narušení soukromí uživatele.

7.2 Kontrola dat

V další fázi vývoje bylo třeba vymyslet kontrolu zachycených dat z předchozího kroku. Ve Wiresharku je možné procházet provoz a v reálném čase si prohlížet jednotlivé poslané pakety. Hlavní myšlenka vyvíjeného nástroje však spočívá v práci na pozadí, aby o něm uživatel nevěděl. Program Tshark se však dá využít pro analýzu zachyceného provozu na pozadí a díky správně sestaveným filtrům tak lze získat informaci, jestli se požadované klíčové slovo vyskytlo (jméno uživatele, port nezabezpečené služby atp.) v zachyceném souboru [20].

Příkaz pro nalezení klíčového slova programem Tshark:

```
tshark -r test.pcap -Y "frame matches pwd" -T fields -e ip.src -e ip.dst -e tcp
    .port
```

Výpis 4: Ukázka filtru pomocí tsharku a výpis IPv4 adres spolu s tcp porty

Výsledek tohoto příkazu je IP adresa zdroje, cílová IP adresa a TCP port:

```
192.168.1.215       205.15.135.8        54360, 80
```

Výpis 5: Výsledek příkazu programu Tshark

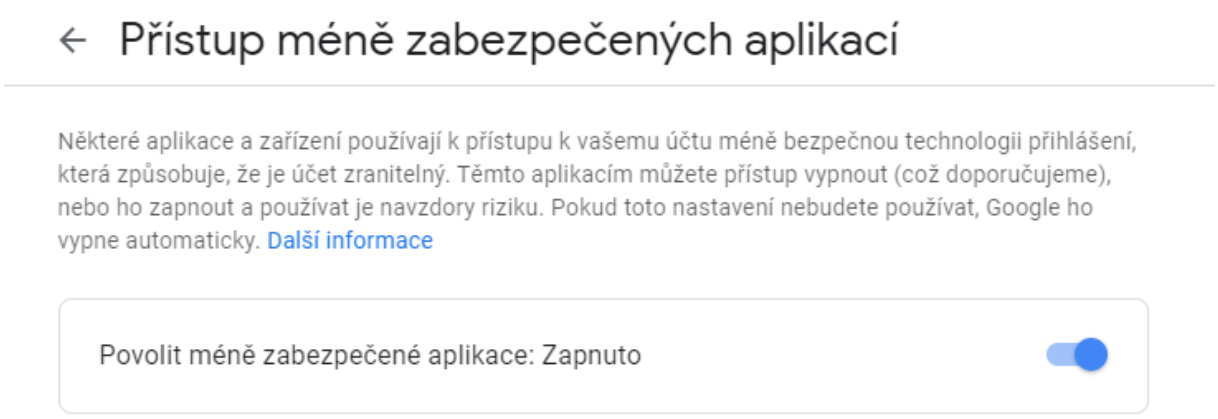
Vyhledávané heslo či uživatelské jméno pak bylo třeba získat od uživatele, ačkoliv pro vyzkoušení stačilo použít heslo v souboru stejné, jako například na stránce, kde se uživatel přihlásí, jenž využívá nezabezpečený protokol HTTP.

7.3 Zasílání emailů

Následně bylo třeba vyřešit, jak informovat uživatele o tom, že došlo k narušení jejich bezpečnosti. Nejvhodnější formou se jevilo zasílání emailu těmto uživatelům. Pro tuto činnost bylo využito knihovny smtplib, kterou programovací jazyk Python nabízí [21].

Knihovna umožňuje zabezpečenou komunikaci s e-mail serverem díky protokolu TLS. Tento protokol se stará o bezpečnou komunikaci mezi klientem a serverem. Jeho primární činností je zabezpečování dat při procházení na internetu, zasílání dat či jiné internetové komunikace. Chrání také proti zneužití dat, které by mohly využít škodlivé aplikace díky naslouchání této komunikace [22].

Aby mohlo zasílaní správně fungovat, je nutné připojit e-mail účet, jenž bude zprávy zasílat. Pro tento případ jsem zvolil Gmail, protože nabízí povolení méně zabezpečeným aplikacím k přístupu na e-mail. Stačí se pouze přihlásit a povolit na stránce přístup jiným aplikacím. Ovšem pro ochranu osobních údajů je lepší si vytvořit nový Gmail účet, který se nepoužívá k osobním účelům. Přihlašovací údaje je také nutno vložit přímo do aplikace, aby se samostatně přihlásila na server a zaslala email [23].



Obrázek 9: Povolení přístupu méně zabezpečeným aplikacím

Poté stačí jen v programu sestavit kostru zprávy a následně ji předat jako parametr funkce knihovny, která zašle email.

Pokud se tedy objeví bezpečnostní riziko, obdrží uživatel email nejpozději do 15 vteřin. Tento časový údaj je dán časovou délkou záznamu sítě. Každých 15 vteřin se kontroluje vytvořený soubor a po této rychlé kontrole se začne vytvářet nový soubor se záznamem. Předchází se tím tak zahlcení složky mnoha soubory najednou nebo vytvořením velkého a nepřehledného souboru. Doba záznamu se dá jednoduše upravit.

7.4 Raspberry Pi

Jde o jednodeskový počítač, který je srovnatelný se (slabším) stolním počítačem. Obsahuje microHDMI porty, USB 2.0/3.0 a Gigabitový ethernet.

Jednodeskový počítač je označení pro malý počítač s jednou deskou plošných spojů. Navzdory malým rozměrům jednodeskových počítačů mívají bohaté možnosti rozšíření o další hardware, jako například vstupně/výstupní moduly. Díky tomu jsou často využívány v projektech IoT [29]. Dokáží samostatně řídit ostatní zařízení, jako například domácí multimediální přehrávač, kamerový systém atp.

K těmto počítačům je možno přistupovat i vzdáleně prostřednictvím programů (VNC), které umožní zobrazit pracovní plochu zařízení nebo pomocí služby SSH.

Pro tuto práci byl použit Raspberry Pi 4 (Model B), který je nástupcem Raspberry Pi 3. Použitý model obsahuje 4 GB RAM. Ta využívá novější typ paměti LPDDR4.

Základním stavebním kamenem Raspberry Pi 4 model B je jeho 1.5 GHz čtyřjádrový procesor ARM Cortex-A72. Přechod na modernější 28nm technologii umožnil Raspberry Pi 4B dosáhnout významného zvýšení výkonu procesoru, multimédií a I/O [30].

Obrázek 10: Raspberry Pi 4 model B

7.5 Nastavení Raspberry Pi jako přístupový bod

Aby se dalo sledovat veškerý provoz na bezdrátové síti pro všechny uživatele, kteří jsou na této síti, z jednoho bodu, bylo nutné nastavit Raspberry Pi na přístupový bod, přes který bude protékat veškerý provoz všech uživatelů na dané síti. Pro tuto část vývoje bylo nutno připojit Raspberry Pi skrze pevný síťový kabel do směrovače internetového poskytovatele [24].

Jako první ze všeho bylo třeba aktualizovat balíčky Raspberry Pi:

```
sudo apt-get update
sudo apt-get upgrade
```

Výpis 6: Aktualizace balíčků

Poté nainstalovat dva balíčky, které umožní nastavení Raspberry Pi na přístupový bod:

```
sudo apt-get install hostapd
sudo apt-get install dnsmasq
```

Výpis 7: Instalace balíčků potřebných k nastavení přístupového bodu

Aby se daly upravit konfigurační soubory těchto služeb, bylo zapotřebí vypnout již nainstalované služby, abychom předešli možným chybám, způsobeným přepisováním konfiguračních souborů těchto služeb:

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Výpis 8: Vypnutí služeb pro správné nastavení

Po zastavení těchto služeb lze upravit konfigurační soubor dhcpcd, a díky němu změnit parametry rozhraní wlan0, přes který protéká veškerý provoz uživatelů v síti:

```
sudo nano /etc/dhcpcd.conf
```

Výpis 9: Otevření konfiguračního souboru pomocí programu nano

Na konci tohoto souboru nastavíme statickou IP adresu rozhraní wlan0:

```
interface wlan0
    static ip_address=192.168.220.1/24
    nohook wpa_supplicant
```

Výpis 10: Nastavení statické IPv4 adresy

Jako další krok bylo nutné restartovat službu dhcpcd pomocí příkazu:

```
sudo systemctl restart dhcpcd
```

Výpis 11: Restart služby dhcpcd

Následně bylo nutné upravit konfigurační soubor služby hostapd, která se stará o vytvoření přístupového bodu na Raspberry Pi:

```
sudo nano /etc/hostapd/hostapd.conf
```

Výpis 12: Otevření konfiguračního souboru služby hostapd

Do tohoto souboru zapíšeme nastavení přístupového bodu:

```
interface=wlan0
driver=nl80211

hw_mode=g
channel=6
ieee80211n=1
wmm_enabled=0
macaddr_acl=0
ignore_broadcast_ssid=0

auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

#Name of your network
ssid=RaspberryAP
#Network password
wpa_passphrase=password123!
```

Výpis 13: Nastavení přístupového bodu na zařízení Raspberry Pi

Uložíme a ukončíme soubor stejně jako v předchozím případě. Dále je zapotřebí upravit ještě dva další soubory, které používá služba hostapd:

```
sudo nano /etc/default/hostapd
```

Výpis 14: Otevření souboru hostapd

V tomto souboru je nutno vyhledat řádek:

```
#DAEMON_CONF=""
```

Výpis 15: Vyhledání zapoznámkovaného řádku

A nahradit ho takto:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Výpis 16: Nahrazení zapoznámkovaného řádku

Následující soubor otevřeme pomocí příkazu:

```
sudo nano /etc/init.d/hostapd
```

Výpis 17: Otevření souboru ve složce obsahující skripty pro práci se službami

Kde je nutno najít řádek:

```
DAEMON_CONF=
```

Výpis 18: Nalezení řádku za účelem následné změny

A upravit jej následovně:

```
DAEMON_CONF=/etc/hostapd/hostapd.conf
```

Výpis 19: Upravení cesty na vyhledaném řádku

Další krok se týká služby dnsmasq. Nejdříve však uložíme původní soubor pro případnou zálohu příkazem:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

Výpis 20: Vytvoření zálohy konfiguračního souboru

Nyní můžeme začít s úpravou nového konfiguračního souboru:

```
sudo nano /etc/dnsmasq.conf
```

Výpis 21: Úprava nově vytvořeného souboru

Do tohoto souboru vložíme následující řádky:

```
interface=wlan0     #Use interface wlan0
server=1.1.1.1      #Use Cloudflare DNS
dhcp-range=192.168.220.50,192.168.220.150,12h #IP range and lease time
```

Výpis 22: Konfigurace souboru dnsmasq

Po uložení souboru přejdeme ke konfiguraci Raspberry Pi. Je potřeba nastvit IPv4 forwarding, kterého docílíme upravením souboru:

```
sudo nano /etc/sysctl.conf
```

Výpis 23: Otevření souboru pro nastavení forwardingu na Raspberry Pi

V tomto souboru najdeme řádek:

```
#net.ipv4.ip_forward=1
```

Výpis 24: Vyhledání řádku pro nastavení forwardingu IPv4

A upravíme následovně:

```
net.ipv4.ip_forward=1
```

Výpis 25: Nastavení forwardingu IPv4

Po uložení lze provést okamžitou aktivaci forwardingu pomocí příkazu:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Výpis 26: Okamžitá aktivace forwardingu

Pro správné fungování přístupového bodu je třeba dále nastavit NAT mezi rozhraním eth0, které je připojeno pomocí kabelu do směrovače poskytovatele a rozhraním wlan0, přes které půjde veškerá bezdrátová komunikace mezi přístupovým bodem Raspberry Pi a ostatními uživateli sítě, kterou tento přístupový bod vytvořil.

K tomu lze použít příkaz:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Výpis 27: Nastavení NAT mezi rozhraním eth0 a wlan0

Při každém načtení systému se však iptables zahodí, proto je nutné zaručit opětovné nahrání iptables, když se systém znovu načte.

Nejdříve tedy uložíme nastavení do souboru:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Výpis 28: Uložení iptables do souboru

Dále upravíme soubor, který se stará o konfiguraci zařízení při spuštění systému:

```
sudo nano /etc/rc.local
```

Výpis 29: Otevření souboru sloužícímu ke konfiguraci zařízení při spuštění systému

Najdeme si v tomto souboru řádek obsahující:

```
exit 0
```

Výpis 30: Požadovaný řádek v konfiguračním souboru

A nad něj vložíme příkaz, pro nahrání nastavení iptables z uloženého souboru:

```
iptables-restore < /etc/iptables.ipv4.nat
```

Výpis 31: Vložení příkazu nad řádek exit 0

Jako poslední můžeme upravit a spustit tyto služby pomocí příkazu systemctl:

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo service dnsmasq start
```

Výpis 32: Úprava a spuštění služeb

Pro ověření funkčnosti můžeme restartovat systém, a poté již bude přístupový bod bez problémů fungovat:

```
sudo reboot
```

Výpis 33: Restart zařízení

7.6 Nastavení Raspberry Pi jako Captive Portal

Captive Portal je jakási uvítací stránka, jež se zobrazí hned po autorizaci do sítě. Aby byli uživatelé této sítě schopni využívat Wi-Fi připojení, musí provést určitou akci v závislosti na nastavení Captive Portalu. Může se jednat například o kliknutí na tlačítko pokračovat nebo se musí uživatel do sítě přihlásit ještě s použitím emailu a uživatelského jména.

Takto lze sledovat, kteří uživatelé v síti jsou a zobrazit si jejich údaje, jako například:

- MAC adresu zařízení
- IP adresu zařízení
- email uživatele
- jméno uživatele
- typ zařízení (iOS, Android)

Aby bylo vůbec možné využívat Captive Portal na Raspberry Pi, bylo nutné mít vytvořený funkční přístupový bod na Raspberry Pi [28] .

Pro omezení vzniku chyb při instalaci a nastavení Captive Portalu, je důležité mít aktualizovaný systém zařízení Raspberry Pi:

```
sudo apt-get update
sudo apt-get upgrade
```

Výpis 34: Aktualizace balíčků

Dále je třeba nainstalovat program git a balíček libmicrohttpd-dev, na kterém závisí správná kompilace kódu aplikace NoDogSplash.

NoDogSplash je Captive Portal, který byl využit pro tuto bakalářskou práci. Jedná se o výkonnou aplikaci, která nabízí jednoduchou základní stránku pro autorizaci uživatelů [25].

Nabízí dvě předinstalované možnosti autorizace:

- Statickou webovou stránku, kde se uživatel autorizuje pouhým kliknutím na tlačítko pokračovat.
- Dynamickou autorizaci pomocí emailu a uživatelského jména uživatele, kdy je nutno zadat oba tyto údaje pro správnou autorizaci a přístup k internetu.

Instalaci programu git a balíčku libmicrohttpd-dev provedeme současně použitím příkazu:

```
sudo apt install git libmicrohttpd-dev
```

Výpis 35: Instalace balíčků git a libmicrohttpd-dev

V dalším kroku je zapotřebí naklonovat NoDogSplash z githubu:

```
cd ~
git clone https://github.com/nodogsplash/nodogsplash.git
```

Výpis 36: Klonování aplikace z githubu v příkazové řádce

Jakmile je dokončeno klonování softwaru, je zapotřebí jej zkompilovat a nainstalovat:

```
cd ~/nodogsplash
make
sudo make install
```

Výpis 37: Kompilace a instalace aplikace NoDogSplash

Po ukončení instalace musíme upravit konfigurační soubor aplikace NoDogSplash:

```
GatewayInterface wlan0
GatewayAddress 192.168.220.1
MaxClients 250
AuthIdleTimeout 480
```

Výpis 38: Upravení konfiguračního souboru NoDogSplash

Přičemž GatewayAddress je IPv4 adresa rozhraní wlan0. Po uložení souboru lze spustit aplikaci NoDogSplash a vyzkoušet funkčnost tak, že provedeme přihlášení do sítě vytvořené přístupovým bodem.

Obrázek 11: Úvodní statická obrazovka NoDogSplash při přihlášení na síť přístupového bodu

Aby se aplikace NoDogSplash spustila při každém spuštění systému, vložíme do souboru /etc/rc.local nad řádek:

```
exit 0
```

Výpis 39: Požadovaný řádek v souboru

Následující příkaz:

```
nodogsplash
```

Výpis 40: Vložení příkazu nad řádek s textem exit 0

Uložíme a při dalším spuštění systému se aplikace NoDogSplash automaticky spustí.

Pro případnou změnu úvodní stránky Captive Portalu je možné využít soubor splash.html, úpravou tohoto souboru docílíme změny vzhledu či funkčnosti základní autorizace uživatelů do sítě.

V rámci bakalářské práce však bylo využito možnosti autorizace prostřednictvím uživatelského jména a emailu, na který je zasláno uživateli upozornění o bezpečnostním riziku na síti.

Aby bylo možné využít tento způsob autorizace, bylo nutno upravit konfigurační soubor aplikace NoDogSplash:

```
sudo nano /etc/nodogsplash/nodogsplash.conf
```

Výpis 41: Otevření konfiguračního souboru aplikace NoDogSplash

V tomto souboru nalezneme řádek:

```
login_option_enabled 0
```

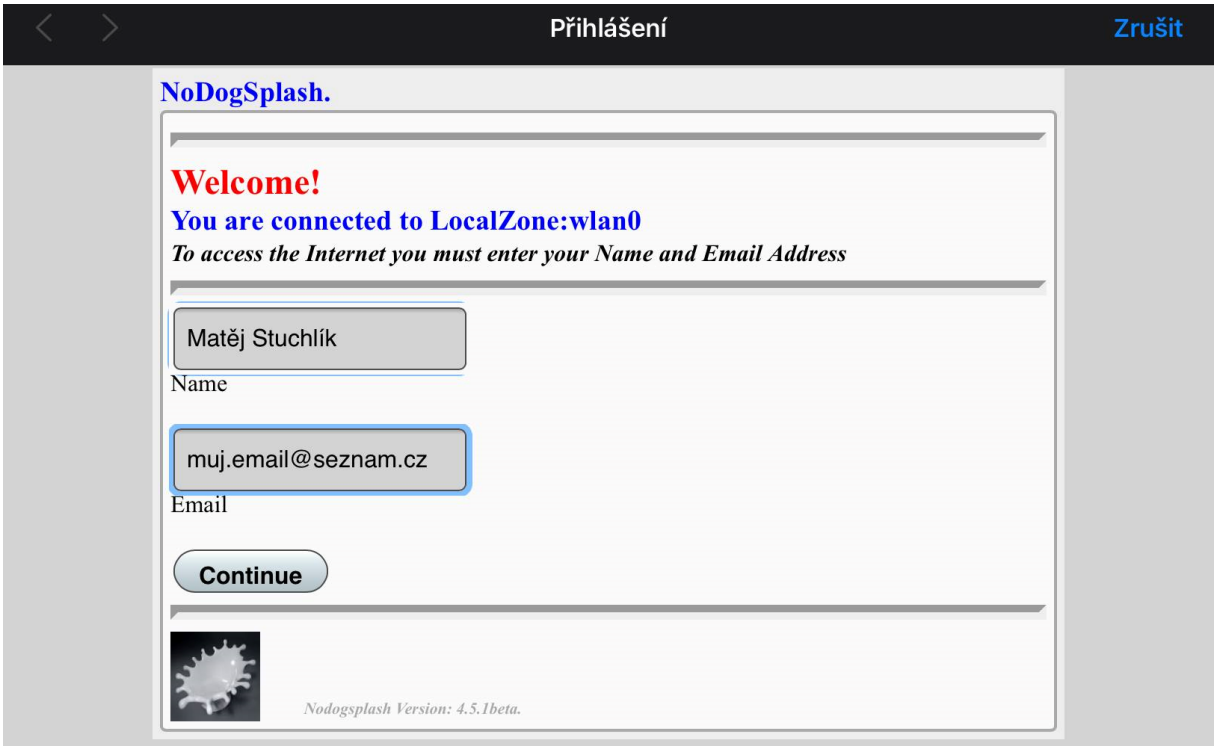
Výpis 42: Nalezení řádku v konfiguračním souboru NoDogSplash

A upravíme následovně, abychom aktivovali možnost autorizace skrze uživatelské jméno a email:

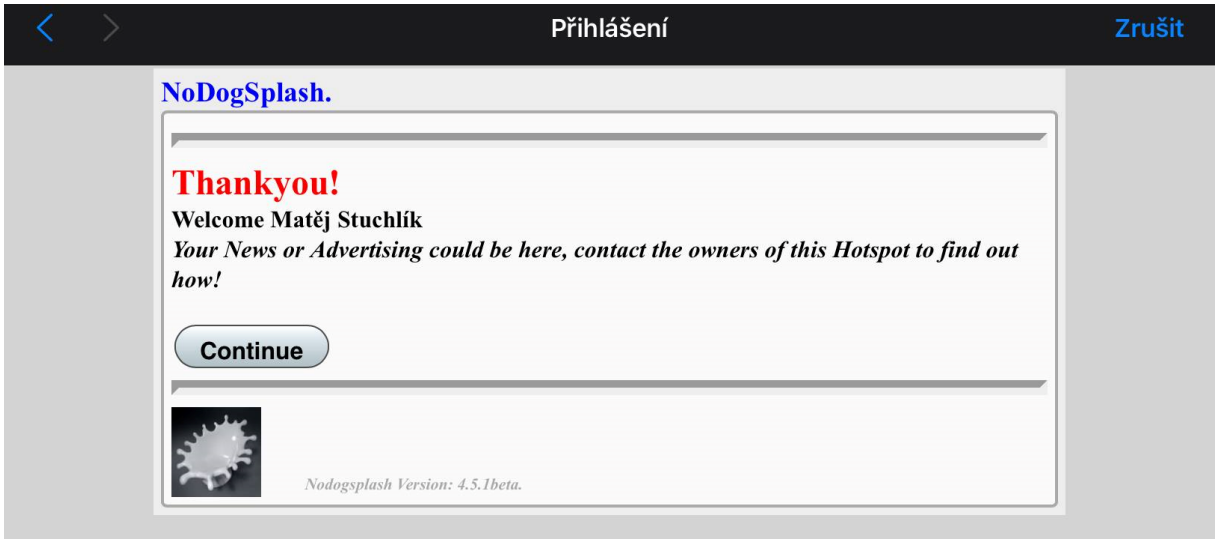
```
login_option_enabled 1
```

Výpis 43: Povolení autorizace na stránce Captive Portalu pomocí emailu a jména

Uložíme a spustíme službu NoDogSplash, abychom mohli otestovat funkčnost autorizace.



Obrázek 12: Přihlášení pomocí autorizace uživatelským jménem a emailem



Obrázek 13: Úspěšná autorizace přes Captive Portal NoDogSplash

8 Návrh testovacího prostředí a testování samotné aplikace

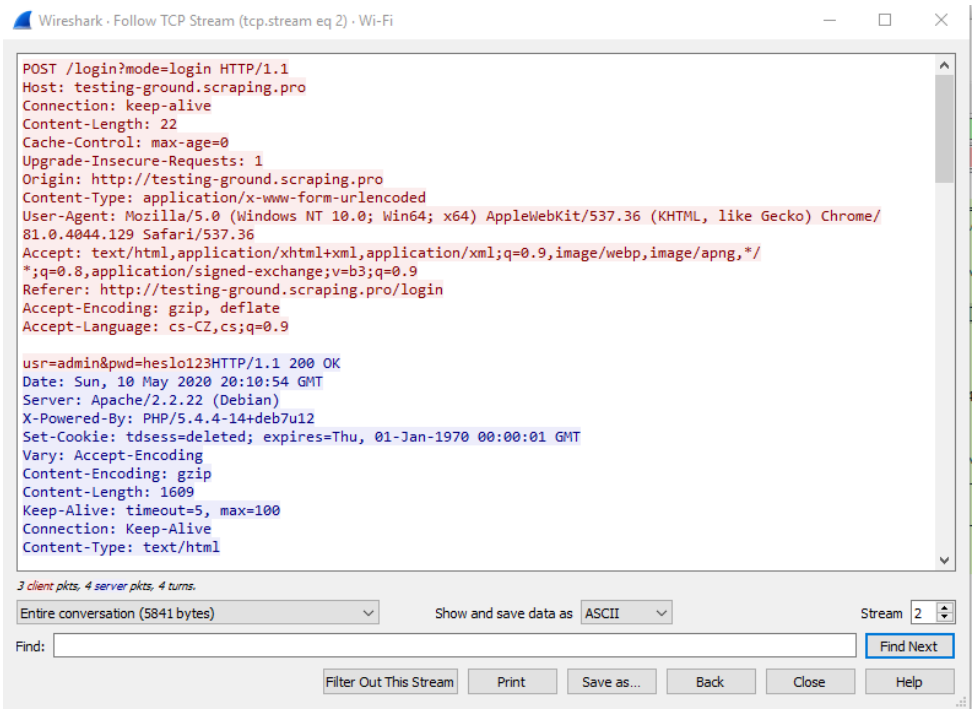
Aplikace síťového provozu kontroluje nezabezpečené služby, které je nutno předem definovat. Může to být například kontrola klíčových slov, jako například pwd pro hesla ve formulářích nebo také čísla portů nezabezpečených služeb (číslo portu 587 pro nezabezpečenou službu serveru pro odchozí poštu SMTP). Je však možno aplikaci jednoduše upravit, aby kontrolovala požadovaná čísla portů, popřípadě další klíčová slova.

8.1 Testování částí aplikace

Samotné testování proběhlo zjištěním funkčností jednotlivých částí. Nejdřív tedy přišlo na řadu monitorování síťového provozu pomocí programu vytvořeného pomocí programovacího jazyka Python s použitím procesů OS Linux, které je možno využít díky Pythonu za použití knihovny os.system. Díky přidání práv pro TCPdump, bylo možné spouštět tyto příkazy bez nutnosti hesla či jinak potřebných práv sudo.

8.2 Zjištění klíčových slov a služeb v záznamu

Dále bylo potřeba zjistit přítomnost citlivých dat uživatele v zachyceném síťovém provozu. První možností bylo vyzkoušet filtr paketů, který je implementován v grafickém rozhraní Wiresharku. Navíc bylo využito funkce prohlédnout si celý TCP Stream paketu, a díky tomu bylo možné zobrazit si, v jakém formátu jsou uvedeny osobní údaje uživatele.

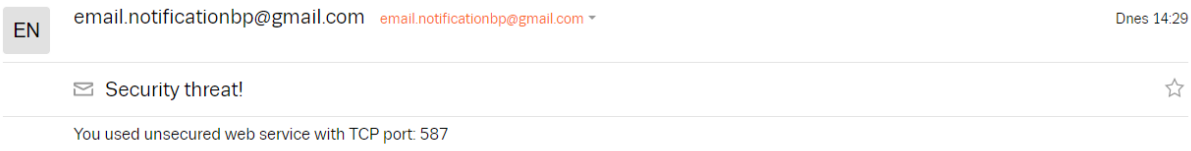


Obrázek 14: Zobrazení TCP Streamu paketu, využívající službu HTTP

Ještě bylo nezbytné vyzkoušet zjištění citlivých údajů pomocí příkazů programu tshark za použití terminálu v OS Linux.

8.3 Zasílání emailů

Tato část testování se věnovala správnému zasílání emailů, které bylo třeba zaslat uživatelům, jenž se přihlásí do aplikace. Aby však bylo otestováno bezchybné odesílání, daly se do kostry zprávy údaje o příjemci staticky přímo v programu.



Obrázek 15: Testovací email s nalezenou službou SMTP

Vše proběhlo bez problémů až po udělení práv aplikaci poskytovatelem emailu, protože pro tuto aplikaci použitý Gmail standardně blokuje všechny aplikace třetích stran z důvodů bezpečnosti osobních údajů a předchází tak jejich zneužití.

8.4 Integrace dat z Captive Portalu do aplikace

Captive Portal obsahuje mnoho užitečných informací o přihlášených uživatelech v logovacím souboru.

```
Thu 07 May 2020 12:30:14 PM CEST, New log file created
Thu 07 May 2020 12:30:14 PM CEST, username=matej, emailAddress=email@uzivatele.
    cz, macaddress=d0:7e:35:d1:bf:c8, clientip=192.168.220.138, clientzone=
    LocalZone:wlan0, useragent=Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36
```

Výpis 44: Výpis logovacího souboru aplikace NoDogSplash

Aby bylo možné získat informace získané prostřednictvím Captive Portalu, bylo zapotřebí upravit přihlašovací skript, který je napsán v BASHi a umístěn v /usr/lib/nodogsplash/login.sh.

```
if [ $sizeratio -ge $min_freespace_to_log_ratio ]; then
      userinfo="username=$username, emailAddress=$emailaddr"
      clientinfo="macaddress=$clientmac, clientip=$clientip, clientzone=
          $client_zone, useragent=$user_agent"
      echo "$datetime, $userinfo, $clientinfo" >> $logfile
      echo "$clientip,$emailaddr" >> /home/pi/Music/testouput.txt
```

Výpis 45: Upravená část přihlašovacího skriptu pro vytvoření souboru s IP a emailem klienta

Tento skript obsahuje mimo jiné i HTML šablonu úvodní stránky a přihlašovací formulář.

```
# Define a login form
login_form="
   <form action=\"/nodogsplash_preauth/\" method=\"get\">
   <input type=\"hidden\" name=\"clientip\" value=\"$clientip\">
   <input type=\"hidden\" name=\"gatewayname\" value=\"$gatewaynamehtml\">
   <input type=\"hidden\" name=\"hid\" value=\"$hid\">
   <input type=\"hidden\" name=\"gatewayaddress\" value=\"$gatewayaddress\">
   <input type=\"hidden\" name=\"redir\" value=\"$requested\">
   <input type=\"text\" name=\"username\" value=\"$usernamehtml\" autocomplete
      =\"on\" ><br>Name<br><br>
   <input type=\"email\" name=\"emailaddr\" value=\"$emailaddr\" autocomplete=\
      "on\" ><br>Email<br><br>
   <input type=\"submit\" value=\"Continue\" >
   </form><hr>
"
```

Výpis 46: Formulář obsažen v přihlašovacím skriptu

8.5 Propojení všech prvků aplikace

Před testováním aplikace v testovacím prostředí bylo třeba spolu propojit všechny části a zaručit plynulý chod aplikace. Bylo nežádoucí, aby se email zaslal svévolně bez jakýchkoliv údajů a zahlcoval tak zbytečně emailovou schránku uživatele, a zároveň nemůže být email zaslán, pokud nedošlo k nalezení osobních údajů, například na webové stránce, nebo využití nezabezpečené služby.

8.6 Testovací prostředí

Jako testovací prostředí byla zvolena domácí síť vytvořená přístupovým bodem Raspberry Pi. Má ochránit běžné uživatele, kteří nejsou nijak informováni o potencionálních hrozbách, které se vyskytují při využívání nezabezpečených webových služeb. Domácí síť používá ochranu WPA2 a obsahuje tak alespoň jakousi základní ochranu. Chtěl jsem tak poukázat na fakt, že i když si mnozí uživatelé myslí, že jejich domácí Wi-Fi síť používá tuto ochranu, může stále dojít k získání jejich dat.

Monitoring se provádí ve smyčce a v případě nalezení rizika se spustí díky rekurzi:

```
def securityMonitoring():
    found = False
    while found is not True:
        capturing.openDump()
        if checkCap.checkIMAP() == 1:
            d = {}
            with open("testouput.txt") as f:
                for line in f:
                    (key, val) = line.split(",")
                    d[str(key)] = val
            print('Vulnerability detected in IMAP service!')
            for key, val in d.items():
                ips = checkCap.getIMAPIPSource().rstrip("\n")
                if key in ips:
                    print("Key and ip.src are equal. Email will be sent.")
                    print("Ports: " + checkCap.getIMAPport())
                    ports=checkCap.getIMAPport()
                    if "143" in ports:
                        sendEmail.sendMailToUserService(val, 143)
                        found = True
                        return securityMonitoring()
```

Výpis 47: Funkce pro zjištění IP adresy uživatele používajícího službu s portem 143.

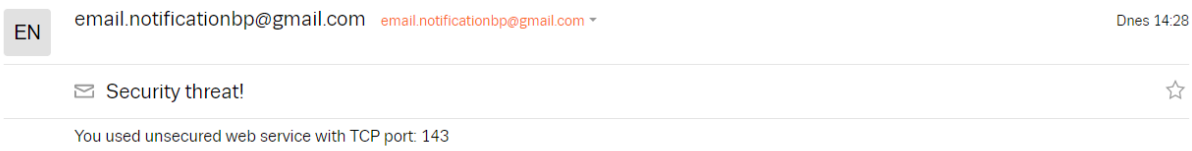
Simuloval jsem tedy uživatele, který se přihlásí na přístupový bod skrze Captive Portal a několik minut surfuje po internetu, následně zašle email přes program, který má nastavený nezabezpečený IMAP server pro příchozí poštu (port 143).

Pokud bude port 143 nalezen a splní tak podmínku v předchozím výpisu kódu:

```
def checkIMAP():
    g = subprocess.Popen('tshark -r /home/pi/Music/output.pcap -Y "tcp.port
        ==143" -T fields -e ip.src', stdout=subprocess.PIPE, shell=True)
    (output, err) = g.communicate()
    if output.decode('utf-8') == "":
        return 0;
    else:
        return 1;
```

Výpis 48: Potvrzení o výskytu nezabezpečeného portu 143

Uživatel využil služby a email mu byl zaslán v následujícím formátu s vypsáním nezabezpečeného portu služby.



Obrázek 16: Obdržený email s číslem portu nezabezpečené služby

9 Vyhodnocení výsledků testů

Testování proběhlo úspěšně a aplikace fungovala podle očekávání. Po spuštění aplikace se průběžně kontrolovali všichni přihlášení uživatelé a jejich IP adresy, a to i uživatelé přihlášení během chodu aplikace. Pokud bylo využito nezabezpečených služeb například (SMTP, IMAP), byl v emailu správně vypsán port nezabezpečené služby a uživatelé tak věděli, o jakou službu se jedná.

Při sledování zátěže na zařízení Raspberry Pi nedošlo během chodu aplikace k významné zátěži jak zařízení, tak i monitorované sítě. Všechny 4 jádra zařízení se držely pod vytížením 15 procent.

Aplikace na Raspberry Pi byla otestována na dvou různých domácích sítích. V obou případech bylo zařízení Raspberry Pi připojeno kabelem k internetu přes rozhraní eth0, bez kterého by nefungoval přístupový bod a aplikace by tedy nebyla schopna zachytit neexistující provoz v důsledku nepřipojení k internetu.

V obou zapojení aplikace neprojevila žádné známky výpadku nebo náhlého vypnutí. Aplikace během testů nenarušila uživatelům využívání internetu, a i během jejího chodu lze využívat samotné zařízení Raspberry Pi bez případných omezení.

10 Závěr

V této bakalářské práci byl vyvinut nástroj pro analýzu nezabezpečeného provozu v bezdrátových sítích, který notifikuje uživatele sítě o případném nebezpečí způsobeným využití nezabezpečených webových služeb. V teoretické části jsou vysvětleny různé pojmy z oblasti bezdrátových sítí a technologie použité k vytvoření aplikace. Práce se zabývala také vytvoření přístupového bodu na zařízení Raspberry Pi 4 Model B a následné nastavení Captive Portalu, což je software umožňující formu autorizace pro uživatele, kteří chtějí využívat internetových služeb sítě vytvořené přístupovým bodem.

Tato aplikace vyžaduje pro svou práci fungující přístupový bod na zařízení Raspberry Pi a vytvořený Captive Portal pro získání IPv4 adresy a emailu přihlášených uživatelů. Zdrojový kód samotné aplikace byl napsán v Pythonu za použití příkazů OS Linux.

Výsledek bakalářské práce je fungující nástroj, který uživatelům umožňuje zvýšit svou ochranu na internetu prostřednictvím zasílání notifikací při možném ohrožení osobních údajů a dat. Veškerý provoz na síti začne být monitorován po přihlášení na přístupový bod a autorizaci skrze Captive Portal.

Téma a samotná práce pro mě představuje obohacení v mnoha oblastech zabezpečení na internetu, a také při vývoji zabezpečovacího softwaru proti případným hrozbám. Abych lépe pochopil princip fungování určitých služeb, musel jsem nahlédnout a prostudovat jejich dokumentaci.

Literatura

[1] *Bezdrátové sítě* [online]. [cit. 2020-04-05]. Dostupné také z: http://www.ped.muni.cz/wtech/old2012/03_studium/teps/teps-07.pdf

[2] ŠEBESTA, Ing. Roman a Ing. Marek DVORSKÝ. *Rádiové sítě I pro integrovanou výuku. Ostrava : Vysoká škola báňská - Technická univerzita Ostrava VUT a VŠB-TUO* [online]. 2014 [cit. 2020-04-05]. Dostupné z: https://lms.vsb.cz/pluginfile.php/990627/mod_resource/content/1/rs1_141022_skripta.pdf

[3] LUCKI, Michal. *Bezdrátové sítě a technologie* [online]. [cit. 2020-04-06]. Dostupné z: http://techpedia.fel.cvut.cz/html/frame.php?oid=50&pid=1003&finf=

[4] ERICKSON, Jon. *Hacking: The Art of Exploitation.* 2nd ed. 555 De Haro Street, Suite 250, San Francisco, CA 94107: No Starch Press, 2008. ISBN 978-1-59327-144-2.

[5] CHAUHAN, Achiv a Jain PALAK. GeeksforGeeks. *TCP/IP Model* [online]. [cit. 2020-04-08]. Dostupné z: https://www.geeksforgeeks.org/tcp-ip-model/

[6] *Wireshark* [online]. [cit. 2020-04-09]. Dostupné z: https://www.wireshark.org/

[7] *Tcpdump and libcap* [online]. [cit. 2020-04-10]. Dostupné z: https://www.tcpdump.org/

[8] *Tshark - Dump and analyze network traffic* [online]. [cit. 2020-04-11]. Dostupné z: https://www.wireshark.org/docs/man-pages/tshark.html

[9] Packet analyzer. *Networx Security* [online]. [cit. 2020-04-12]. Dostupné z: https://www.networxsecurity.org/members-area/glossary/p/packet-sniffing.html

[10] Monitor mode. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-04-13]. Dostupné z: https://en.wikipedia.org/wiki/Monitor_mode

[11] TOURRHILHES, Jean. *Iwconfig - Linux man page* [online]. [cit. 2020-04-13]. Dostupné z: https://linux.die.net/man/8/iwconfig

[12] DHAR, Sumit. *Sniffers - Basics and Detection* [online]. [cit. 2020-04-15]. Dostupné také z: http://www.just.edu.jo/~tawalbeh/nyit/incs745/presentations/Sniffers.pdf

[13] *Jak na internet* [online]. CZ.NIC, 2012- [cit. 2020-04-16]. Dostupné z: https://www.jaknainternet.cz/page/1251/sifrovani/

[14] LASHKARI H., A., MANSOOR a DANESH. IEEEXplore. *Wired Equivalent Privacy (WEP) versus Wi-Fi Protected Access (WPA)* [online]. 2009- [cit. 2020-04-18]. Dostupné z: https://ieeexplore.ieee.org/document/5166826

[15] Jaký je rozdíl mezi WPA2, WPA3, WPA, WEP, AES a TKIP?. *Airwaynet* [online]. [cit. 2020-04-19]. Dostupné z: https://www.airwaynet.cz/jaky-je-rozdil-mezi-wpa2-wpa3-wpa-wep-aes-a-tkip/

[16] *Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks* [online]. 2003- [cit. 2020-04-19]. Dostupné také z: http://kambing.ui.ac.id/onnopurbo/library/library-ref-eng/ref-eng-3/physical/wireless/security/wp_8_WPA%20Security_4-29-03.pdf

[17] Wireless Security Protocols: WEP, WPA, WPA2, and WPA3. *Netspotapp* [online]. Netspot, [cit. 2020-04-20]. Dostupné z: https://www.netspotapp.com/wifi-encryption-and-security.html

[18] Subprocess management. *Python dokumentace* [online]. [cit. 2020-04-20]. Dostupné z: https://docs.python.org/3/library/subprocess.html

[19] Python sleep(). *Programiz* [online]. [cit. 2020-04-22]. Dostupné z: https://www.programiz.com/python-programming/time/sleep

[20] Tshark tutorial and filter examples. *Hacker Target* [online]. 2015- [cit. 2020-04-22]. Dostupné z: https://hackertarget.com/tshark-tutorial-and-filter-examples

[21] SMTP protocol client. *Python dokumentace* [online]. [cit. 2020-04-23]. Dostupné z: https://docs.python.org/3/library/smtplib.html

[22] Transport Layer Security (TLS). *Techopedia* [online]. [cit. 2020-04-25]. Dostupné z: https://www.techopedia.com/definition/4143/transport-layer-security-tls

[23] Přístup méně zabezpečených aplikací. *GMail* [online]. [cit. 2020-04-25]. Dostupné z: https://myaccount.google.com/lesssecureapps

[24] Raspberry Pi Wireless Access Point. *PiMyLifeUp* [online]. 2017- [cit. 2020-04-26]. Dostupné z: https://pimylifeup.com/raspberry-pi-wireless-access-point/

[25] NoDogSplash. *GitHub* [online]. [cit. 2020-04-29]. Dostupné z: https://github.com/nodogsplash/nodogsplash

[26] WiGig: IEEE 802.11ad 60GHz Microwave Wi-Fi. *Electronics-notes* [online]. [cit. 2020-04-30]. Dostupné z: https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/802-11ad-wigig-gigabit-microwave.php

[27] Why change your Wi-Fi from WPA2 to WPA3?. *Kyos* [online]. [cit. 2020-04-18]. Dostupné z: https://channelworld.cz/software/k-cemu-je-dobre-wpa3-21474

[28] Setting up a Raspberry Pi Captive Portal. *PiMyLifeUp* [online]. 2017- [cit. 2020-04-26]. Dostupné z: https://pimylifeup.com/raspberry-pi-captive-portal/

[29] Jednodeskový počítač. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-04-14]. Dostupné z: https://cs.wikipedia.org/wiki/Jednodeskov%C3%BD_po%C4%8D%C3%ADta%C4%8D

[30] Raspberry Pi 4. *Raspberry Pi Foundation* [online]. [cit. 2020-04-29]. Dostupné z: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

A Zdrojové kódy

security.py – spouštěcí soubor

monitoring.py – soubor, ve kterém se nachází hlavní smyčka monitorování

capturing.py – soubor pro zachytávání provozu na síti

checkCap.py – nalézání konkrétních portů a klíčových slov v zachyceném provozu

sendEmail.py – zasílání emailů uživatelům

B Konfigurační soubory a skripty

dnsmasq.conf - nastavení DNS serveru a rozsah DHCP

hostapd.conf - konfigurace přístupového bodu

login.sh - přihlašovací skript autorizačního přístupu jména a emailu

nodogsplash.conf - konfigurace aplikace Captive Portalu